A TWO STEP OPTIMIZATION BASED ITERATIVE LEARNING CONTROL ALGORITHM

Marnix Volckaert[*], **Jan Swevers** **Moritz Diehl**
Department of Mechanical Engineering Department of Electrical Engineering
Katholieke Universiteit Leuven
Leuven, Belgium
Email: marnix.volckaert@mech.kuleuven.be

ABSTRACT

This article introduces a general formulation of model based iterative learning control (ILC). The formulation is valid for both linear and nonlinear systems. It is a two step approach, such that after each repetition of the task two (non)linear least squares problems have to be solved. In the first step an optimal model correction is calculated. This is a nonparametric correction to the model in order to describe the measured output signal more accurately. This model correction is used in the second step, which is a model inversion problem. Conventional linear ILC is shown to be a particular case of this general formulation.

INTRODUCTION

The purpose of the research presented in this article is to accurately control systems that perform a given task repeatedly. The task is represented by a reference output $\mathbf{y}_r$, that needs to be followed by the system's output $\mathbf{y}$. Iterative learning control is an open loop control strategy that exploits the repetition of the task, to iteratively improve the input signal $\mathbf{u}$ that is applied to the system.

The first publication in english on ILC is attributed to Arimoto *et al.* [1], in which the most basic input learning law is formulated as follows:

$$\mathbf{u}^{i+1} = \mathbf{u}^i + \Gamma \mathbf{e}^i. \tag{1}$$

In this equation $\mathbf{u}^i$ is the input signal of the i^{th} iteration, Γ is a constant scalar, and $\mathbf{e}^i$ is the tracking error, defined as $\mathbf{e}^i = \mathbf{y}_r - \mathbf{y}^i$. Note that $\mathbf{u}$ and $\mathbf{y}$ are vectors, defined as $\mathbf{u} = [u(0)\ u(1)\ \cdots\ u(N-1)]^{\mathrm{T}}$ and $\mathbf{y} = [y(m)\ y(m+1)\ \cdots\ y(N-1+m)]^{\mathrm{T}}$ with N the number of samples in the signal, and m the relative degree of the system.

The scalar Γ determines the relation between the tracking error of the present iteration, and the update to the input signal of the next iteration. Several ILC algorithms formulate this relation based on a model of the system, and are therefore called model based ILC. In this paper, a model is represented by $\mathbf{y} = \hat{P}(\mathbf{u})$, while a plant is represented by $\mathbf{y} = P(\mathbf{u})$. The most common model based linear ILC algorithm, described in detail in [2], is formulated as follows:

$$\mathbf{u}^{i+1} = Q[\mathbf{u}^i + L(\mathbf{e}^i)], \tag{2}$$

with $Q(\cdot)$ and $L(\cdot)$ a robustness and learning operator, respectively. Since $\mathbf{u}$ and $\mathbf{e}$ are vectors, it is convenient to write these linear operators in matrix form. For example, consider a discrete-time LTI system with relative degree 1, characterised by a set of Markov parameters $p_1, p_2, \ldots$. The input-output relation of this system can be written as:

$$\begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix} = \begin{bmatrix} y_0(1) \\ y_0(2) \\ \vdots \\ y_0(N) \end{bmatrix} + \underbrace{\begin{bmatrix} p_1 & 0 & \cdots & 0 \\ p_2 & p_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ p_N & p_{N-1} & \cdots & p_1 \end{bmatrix}}_{\mathbf{P}} \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix}, \tag{3}$$

with $\mathbf{y}_0 = [(CA)^{\mathrm{T}}\ (CA^2)^{\mathrm{T}}\ \cdots\ (CA^N)^{\mathrm{T}}]^{\mathrm{T}}\mathbf{x}_0$, and $\mathbf{x}_0$ the initial state of the system. It can be assumed without loss of generality that

[*]Address all correspondence to this author. This work benefits from FWO-project G.0422.08, K.U.Leuven-BOF EF/05/006 Center-of-Excellence Optimization in Engineering and the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office (DYSCO).

$\mathbf{x}_0 = 0$, such that $\mathbf{y} = \mathbf{P}\mathbf{u}$. Note that in this case the plant operator is $P(\mathbf{u}) = \mathbf{P}\mathbf{u}$, and $Q(\cdot)$ and $L(\cdot)$ can be expressed by similar matrices $\mathbf{Q}$ and $\mathbf{L}$ of their Markov parameters. It is assumed that the plant is modelled by a matrix $\hat{\mathbf{P}}$, so $\hat{P}(\mathbf{u}) = \hat{\mathbf{P}}\mathbf{u}$ and $\hat{\mathbf{P}} \approx \mathbf{P}$. ILC law (2) can then be designed to minimize a next iteration cost function [2], typically of the form:

$$J^{i+1}(\mathbf{u}^{i+1}) = \mathbf{e}^{i+1\mathrm{T}}\mathbf{Q}_\mathbf{e}\mathbf{e}^{i+1} + \mathbf{u}^{i+1\mathrm{T}}\mathbf{R}_\mathbf{u}\mathbf{u}^{i+1}, \tag{4}$$

where $\mathbf{Q}_\mathbf{e}$ is an $N \times N$ positive-definite matrix, and $\mathbf{R}_\mathbf{u}$ is an $N \times N$ positive-semidefinite matrix. This approach is often called quadratically optimal ILC, and will be called conventional ILC in this paper. The first term of this cost function depends on the unknown next iteration tracking error $\mathbf{e}^{i+1}$. If the change in $\mathbf{u}$ from iteration i to $i+1$ is denoted by δ^i, then $\mathbf{u}^{i+1} = \mathbf{u}^i + \delta^i$, and $\mathbf{y}^{i+1} = \mathbf{y}^i + \mathbf{P}\delta^i$. The true expression for $\mathbf{e}^{i+1}$ can therefore be formulated as $\mathbf{e}^{i+1} = \mathbf{y}_r - \mathbf{y}^i - \mathbf{P}\delta^i$.

In order to minimize Eq. (4), the conventional ILC approximates this expression using the model $\hat{\mathbf{P}}$ instead of $\mathbf{P}$, so $\mathbf{e}^{i+1} \approx \mathbf{y}_r - \mathbf{y}^i - \hat{\mathbf{P}}\delta^i$. The ILC law that minimizes Eq. (4), where $\mathbf{e}^{i+1}$ is approximated by the given expression, equals [2]:

$$\mathbf{u}^{i+1} = (\hat{\mathbf{P}}^\mathrm{T}\mathbf{Q}_\mathbf{e}\hat{\mathbf{P}} + \mathbf{R}_\mathbf{u})^{-1}\hat{\mathbf{P}}^\mathrm{T}\mathbf{Q}_\mathbf{e} \cdot \Big[\mathbf{y}_r - (\mathbf{y}^i - \hat{\mathbf{P}}\mathbf{u}^i)\Big]. \tag{5}$$

This corresponds to Eq. (2) with [2]:

$$\begin{aligned} \mathbf{Q} &= (\hat{\mathbf{P}}^\mathrm{T}\mathbf{Q}_\mathbf{e}\hat{\mathbf{P}} + \mathbf{R}_\mathbf{u})^{-1}(\hat{\mathbf{P}}^\mathrm{T}\mathbf{Q}_\mathbf{e}\hat{\mathbf{P}}), \\ \mathbf{L} &= (\hat{\mathbf{P}}^\mathrm{T}\mathbf{Q}_\mathbf{e}\hat{\mathbf{P}})^{-1}\hat{\mathbf{P}}^\mathrm{T}\mathbf{Q}_\mathbf{e}. \end{aligned} \tag{6}$$

Note that if $\mathbf{Q}_\mathbf{e} = I^{N\times N}$ and $\mathbf{R}_\mathbf{u} = 0^{N\times N}$, the optimal operators are found to be $\mathbf{Q} = I^{N\times N}$ and $\mathbf{L} = \hat{\mathbf{P}}^{-1}$, which is the simplest form of model based linear ILC.

Although Eq. (4) formulates a general objective for all ILC algorithms (reducing the tracking error in the next iteration), the solution method gives rise to a number of limitations on the resulting algorithm. First, since the optimal solution is not calculated directly but captured in the operators $\mathbf{Q}$ and $\mathbf{L}$, there is no possibility of including constraints in the solution, such as input constraints to avoid saturation. Second, these expressions (6) of $\mathbf{Q}$ and $\mathbf{L}$ are only valid for LTI systems.

Other ILC approaches have been developed based on the same cost function with various optimization algorithms. For example, [3] and [4] describe the norm-optimal ILC algorithm, which is found by indirectly solving the optimization problem as an LQR problem. This algorithm is subject to the same limitations as the conventional ILC, and furthermore requires full state knowledge during operation.

It is therefore desirable to design ILC algorithms that minimize the same cost function (4), and in addition can account for constraints and are applicable to linear and nonlinear systems. This paper presents such an ILC algorithm.

The next section introduces the two step approach as a general formulation of ILC, and shows that the conventional ILC is a particular case of it. This is illustrated in the third section by a numerical example, followed by conclusions.

TWO STEP APPROACH

This section introduces the developed two step approach. It was shown in the previous section that the conventional ILC approximates the next iteration tracking error by $\mathbf{e}^{i+1} \approx \mathbf{y}_r - \mathbf{y}^i - \hat{\mathbf{P}}\delta^i$. Substituting δ^i by $\mathbf{u}^{i+1} - \mathbf{u}^i$ in this expression yields:

$$\mathbf{e}^{i+1} \approx \mathbf{y}_r - \hat{\mathbf{P}}\mathbf{u}^{i+1} - (\mathbf{y}^i - \hat{\mathbf{P}}\mathbf{u}^i). \tag{7}$$

Equation (7) can be interpreted as follows: $\mathbf{y}_r - \hat{\mathbf{P}}\mathbf{u}^{i+1}$ is the best approximation of the next iteration tracking error for the available model $\hat{\mathbf{P}}$, and $(\mathbf{y}^i - \hat{\mathbf{P}}\mathbf{u}^i)$ is a correction term used to improve the tracking error approximation. Therefore it is clear that the conventional ILC implicitly corrects the model after each iteration, by assuming $\mathbf{e}^{i+1} \approx \mathbf{y}_r - \hat{P}_c(\mathbf{u}^{i+1})$ with a corrected model $\hat{P}_c(\mathbf{u}^{i+1}) = \hat{\mathbf{P}}\mathbf{u}^{i+1} + (\mathbf{y}^i - \hat{\mathbf{P}}\mathbf{u}^i)$.

The two step approach aims at making this model correction explicit, in order to increase the algorithm's flexibility compared to the conventional ILC, for example by allowing different forms of correction. If the correction term is written as a vector α, a corrected model can be derived in several ways, for example:

$$\hat{P}_c(\mathbf{u},\alpha) = \quad \hat{P}(\mathbf{u}) + \alpha, \quad \hat{P}(\mathbf{u}+\alpha), \quad \mathrm{diag}(\alpha)\cdot\hat{P}(\mathbf{u}). \tag{8}$$

In the general case $\hat{P}(\mathbf{u})$ can be either a linear or a nonlinear model. Note that the conventional ILC uses a correction in the form of the first term of Eq. (8), with $\alpha = \mathbf{y} - \hat{\mathbf{P}}\mathbf{u}$.

The first step of the two step approach now consists of calculating the optimal value of α^i after each iteration, such that $\hat{P}_c(\mathbf{u}^\mathbf{i}, \alpha^\mathbf{i})$ describes $\mathbf{y}^\mathbf{i}$ better than $\hat{P}(\mathbf{u}^\mathbf{i})$. The optimization problem that constitutes this first step can be formulated as:

$$\begin{aligned} \alpha^i = \arg\min_\alpha \ & \|\mathbf{y}^i - \hat{P}_c(\mathbf{u}^i,\alpha)\|^2_{\mathbf{Q}_\alpha} + \|\alpha\|^2_{\mathbf{R}_\alpha} + \|\alpha - \alpha^{i-1}\|^2_{\mathbf{S}_\alpha} \\ & \text{s.t.} \\ & g_\alpha(\alpha) \leq 0, \end{aligned} \tag{9}$$

where $\|\cdot\|$ denotes the weighted 2-norm, $\mathbf{Q}_\alpha$ is an $N \times N$ positive-definite matrix, and $\mathbf{R}_\alpha$ and $\mathbf{S}_\alpha$ are $N \times N$ positive-semidefinite matrices. The function $g_\alpha(\alpha)$ is a (non)linear function to constrain the vector α. Note that if $\hat{P}(\mathbf{u})$ is a linear model, $\mathbf{Q}_\alpha = I^{N\times N}$, $\mathbf{R}_\alpha = \mathbf{S}_\alpha = 0^{N\times N}$ and $g_\alpha(\alpha) = 0$, the solution of Eq. (9) is $\alpha = \mathbf{y}^i - \hat{\mathbf{P}}\mathbf{u}^i$. This means that the quadratically optimal ILC is a particular case of the more general formulation of Eq. (9).

Regularization with $\mathbf{R}_\alpha \neq 0^{N\times N}$ allows control over the learning behavior of the algorithm, in time domain or frequency domain. For example, increasing the weighting of this term at

specific samples reduces the learning only at this part of the signal, which can avoid local instabilities of the learning algorithm. $\mathbf{S}_\alpha \neq 0^{N\times N}$ introduces a memory in the learning dynamics, similar to higher order ILC algorithms. The application of inequality constraints allows to incorporate knowledge of the model quality. Limiting the model correction at particular parts of the signal can improve learning stability, at the cost of optimization freedom.

Solving Eq. (9) provides a corrected model $\hat{P}_c(\mathbf{u},\alpha)$ that can be used to estimate the next iteration tracking error $\mathbf{e}^{i+1}$, as $\mathbf{e}^{i+1} \approx \mathbf{y}_r - \hat{P}_c(\mathbf{u}^{i+1},\alpha^i)$.

The second step of the two step approach is then to calculate the optimal next iteration input signal $\mathbf{u}^{i+1}$. Optimality in this case is defined as to minimize a cost function of the form of Eq. (4), augmented with inequality constraints. The second step can therefore be formally written as:

$$\mathbf{u}^{i+1} = \arg\min_{\mathbf{u}} \|\mathbf{y}_r - \hat{P}_c(\mathbf{u},\alpha^i)\|^2_{\mathbf{Q_u}} + \|\mathbf{u}\|^2_{\mathbf{R_u}} \qquad (10)$$
$$\text{s.t.}$$
$$g_{\mathbf{u}}(\mathbf{u}) \leq 0,$$

with $\mathbf{Q_u}$ an $N\times N$ positive-definite matrix, and $\mathbf{R_u}$ an $N\times N$ positive-semidefinite matrix. The application of the regularization term on $\mathbf{u}^{i+1}$ with $\mathbf{R_u} \neq 0^{N\times N}$ can be necessary to apply the algorithm to non-minimum phase systems [5], while constraints of the form $g_{\mathbf{u}}(\mathbf{u}) \leq 0$ allow the algorithm to prevent saturation.

The optimization problems of both step one and two are least squares problems, linear or nonlinear, depending on the linearity of the model $\hat{P}(u)$, on the choice of model correction, and on the presence of constraints. Such problems can be solved efficiently with a direct approach, such as a constrained Gauss-Newton method.

NUMERICAL EXAMPLE

In order to illustrate the fact that the conventional ILC is a specific case of the general two step approach, a comparison is made between both methods in a numerical example. The plant $P(\mathbf{u})$ is assumed to be a mass-spring-damper system with a resonance frequency of 2 rad/s and a damping ratio of 0.28, while the available linear model $\hat{P}(\mathbf{u})$ is assumed to have a resonance frequency of 1.6 rad/s and a damping ratio of 0.34, in order to create a model-plant mismatch. Model correction is assumed to be of the form of the first term in Eq. (8), and the model correction step is carried out with $\mathbf{Q}_\alpha = I^{N\times N}$ and $\mathbf{R}_\alpha = \mathbf{S}_\alpha = 0^{N\times N}$. The second step is carried out with $\mathbf{Q_u} = I^{N\times N}$ and $\mathbf{R_u}$ a matrix that penalizes the derivative of $\mathbf{u}$. The quadratically optimal ILC is constructed according to (6) with the same weighting matrices. Figure 1 shows the result of both ILC approaches. It is clear that both the conventional ILC and the two step approach have the same performance and lead to the same results.

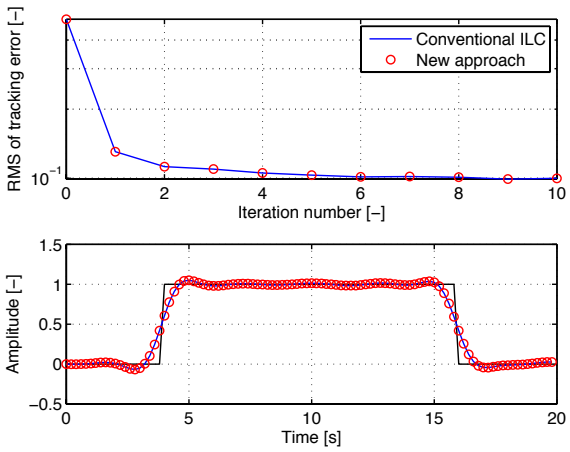

Figure 1. Convergence of the tracking error (top) and final tracking performance in time domain (bottom) of conventional ILC and the new two step approach

CONCLUSIONS

This article has presented a general iterative learning control approach, consisting of two steps: a model correction step, followed by a model inversion step. The method is general in the sense that it can be applied to linear and nonlinear systems. Both the model correction and the model inversion are (non)linear least squares problems, which can be efficiently solved using a constrained Gauss-Newton algorithm. The two step approach is flexible, since the application of regularization terms and inequality constraints allows maximal control over the behavior of the learning algorithm. It is shown by a numerical example that the conventional ILC is a particular case of the general approach.

REFERENCES

[1] Arimoto, S., Kawamura, S., and Miyazaki, F., 1984. "Bettering operations of robots by learning". *Journal of Robotic Systems,* **1**, pp. 123–140.

[2] Bristow, D. A., Tharayil, M., and Alleyne, A. G., 2006. "A survey of iterative learning control: A learning-based method for high-performance tracking control". *IEEE Control Systems Magazine,* **26**(3), pp. 96–114.

[3] Amann, N., Owens, D., and Rogers, E., 1996. "Iterative learning control for discrete time systems with exponential rates of rate convergence". *Proceeding of The Institution of Electrical Engineers on Control Theory and Applications,* **143**(2), pp. 217–224. accepted for publication.

[4] Amann, N., Owens, D. H., and Rogers, E., 1998. "Predictive optimal iterative learning control". *International Journal of Control,* **69**(2), pp. 203–226.

[5] Gunnarsson, S., and Norrlof, M., 2001. "On the design of ilc algorithms using optimization". *Automatica,* **37**(12), Jan, pp. 2011–2016.